Postel

Network Working Group
Request for Comments: 1030

M. Lambert
M.I.T. Laboratory for Computer Science
November 1987

On Testing the NETBLT Protocol over Divers Networks

STATUS OF THIS MEMO

This RFC describes the results gathered from testing NETBLT over three networks of differing bandwidths and round-trip delays. While the results are not complete, the information gathered so far has been very promising and supports RFC-998's assertion that that NETBLT can provide very high throughput over networks with very different characteristics. Distribution of this memo is unlimited.

1. Introduction

NETBLT (NETwork BLock Transfer) is a transport level protocol intended for the rapid transfer of a large quantity of data between computers. It provides a transfer that is reliable and flow controlled, and is designed to provide maximum throughput over a wide variety of networks. The NETBLT protocol is specified in RFC-998; this document assumes an understanding of the specification as described in RFC-998.

Tests over three different networks are described in this document. The first network, a 10 megabit-per-second Proteon Token Ring, served as a "reference environment" to determine NETBLT's best possible performance. The second network, a 10 megabit-per-second Ethernet, served as an access path to the third network, the 3 megabit-per-second Wideband satellite network. Determining NETBLT's performance over the Ethernet allowed us to account for Ethernet-caused behaviour in NETBLT transfers that used the Wideband network. Test results for each network are described in separate sections. The final section presents some conclusions and further directions of research. The document's appendices list test results in detail.

2. Acknowledgements

Many thanks are due Bob Braden, Stephen Casner, and Annette DeSchon of ISI for the time they spent analyzing and commenting on test results gathered at the ISI end of the NETBLT Wideband network tests. Bob Braden was also responsible for porting the IBM PC/AT NETBLT implementation to a SUN-3 workstation running UNIX. Thanks are also due Mike Brescia, Steven Storch, Claudio Topolcic and others at BBN who provided much useful information about the Wideband network, and

helped monitor it during testing.

3. Implementations and Test Programs

This section briefly describes the NETBLT implementations and test programs used in the testing. Currently, NETBLT runs on three machine types: Symbolics LISP machines, IBM PC/ATs, and SUN-3s. The test results described in this paper were gathered using the IBM PC/AT and SUN-3 NETBLT implementations. The IBM and SUN implementations are very similar; most differences lie in timer and multi-tasking library implementations. The SUN NETBLT implementation uses UNIX's user-accessible raw IP socket; it is not implemented in the UNIX kernel.

The test application performs a simple memory-to-memory transfer of an arbitrary amount of data. All data are actually allocated by the application, given to the protocol layer, and copied into NETBLT packets. The results are therefore fairly realistic and, with appropriately large amounts of buffering, could be attained by disk-based applications as well.

The test application provides several parameters that can be varied to alter NETBLT's performance characteristics. The most important of these parameters are:

burst interval The number of milliseconds from the start of one burst transmission to the start of the next burst transmission.

burst size The number of packets transmitted per burst.

buffer size The number of bytes in a NETBLT buffer (all buffers must be the same size, save the last, which can be any size required to complete the transfer).

data packet size
The number of bytes contained in a NETBLT DATA packet's data segment.

number of outstanding buffers
The number of buffers which can be in transmission/error recovery at any given moment.

The protocol's throughput is measured in two ways. First, the "real throughput" is throughput as viewed by the user: the number of bits transferred divided by the time from program start to program finish. Although this is a useful measurement from the user's point of view, another throughput measurement is more useful for analyzing NETBLT's performance. The "steady-state throughput" is the rate at which data is transmitted as the transfer size approaches infinity. It does not take into account connection setup time, and (more importantly), does not take into account the time spent recovering from packet-loss errors that occur after the last buffer in the transmission is sent out. For NETBLT transfers using networks with long round-trip delays (and consequently with large numbers of outstanding buffers), this "late" recovery phase can add large amounts of time to the transmission, time which does not reflect NETBLT's peak transmission rate. The throughputs listed in the test cases that follow are all steady-state throughputs.

4. Implementation Performance

This section describes the theoretical performance of the IBM PC/AT NETBLT implementation on both the transmitting and receiving sides. Theoretical performance was measured on two LANs: a 10 megabit-per-second Proteon Token Ring and a 10 megabit-per-second Ethernet. "Theoretical performance" is defined to be the performance achieved if the sending NETBLT did nothing but transmit data packets, and the receiving NETBLT did nothing but receive data packets.

Measuring the send-side's theoretical performance is fairly easy, since the sending NETBLT does very little more than transmit packets at a predetermined rate. There are few, if any, factors which can influence the processing speed one way or another.

Using a Proteon P1300 interface on a Proteon Token Ring, the IBM PC/AT NETBLT implementation can copy a maximum-sized packet (1990 bytes excluding protocol headers) from NETBLT buffer to NETBLT data packet, format the packet header, and transmit the packet onto the network in about 8 milliseconds. This translates to a maximum theoretical throughput of 1.99 megabits per second.

Using a 3COM 3C500 interface on an Ethernet LAN, the same implementation can transmit a maximum-sized packet (1438 bytes excluding protocol headers) in 6.0 milliseconds, for a maximum theoretical throughput of 1.92 megabits per second.

Measuring the receive-side's theoretical performance is more difficult. Since all timer management and message ACK overhead is incurred at the receiving NETBLT's end, the processing speed can be slightly slower than the sending NETBLT's processing speed (this does

not even take into account the demultiplexing overhead that the receiver incurs while matching packets with protocol handling functions and connections). In fact, the amount by which the two processing speeds differ is dependent on several factors, the most important of which are: length of the NETBLT buffer list, the number of data timers which may need to be set, and the number of control messages which are ACKed by the data packet. Almost all of this added overhead is directly related to the number of outstanding buffers allowable during the transfer. The fewer the number of outstanding buffers, the shorter the NETBLT buffer list, and the faster a scan through the buffer list and the shorter the list of unacknowledged control messages.

Assuming a single-outstanding-buffer transfer, the receiving-side NETBLT can DMA a maximum-sized data packet from the Proteon Token Ring into its network interface, copy it from the interface into a packet buffer and finally copy the packet into the correct NETBLT buffer in 8 milliseconds: the same speed as the sender of data.

Under the same conditions, the implementation can receive a maximum-sized packet from the Ethernet in 6.1 milliseconds, for a maximum theoretical throughput of 1.89 megabits per second.

5. Testing on a Proteon Token Ring

The Proteon Token Ring used for testing is a 10 megabit-per-second LAN supporting about 40 hosts. The machines on either end of the transfer were IBM PC/ATs using Proteon P1300 network interfaces. The Token Ring provides high bandwidth with low round-trip delay and negligible packet loss, a good debugging environment in situations where packet loss, packet reordering, and long round-trip time would hinder debugging. Also contributing to high performance is the large (maximum 2046 bytes) network MTU. The larger packets take somewhat longer to transmit than do smaller packets (8 milliseconds per 2046 byte packet versus 6 milliseconds per 1500 byte packet), but the lessened per-byte computational overhead increases throughput somewhat.

The fastest single-outstanding-buffer transmission rate was 1.49 megabits per second, and was achieved using a test case with the following parameters:

transfer size	2-5 million bytes
data packet size	1990 bytes
buffer size	19900 bytes
burst size	5 packets
burst interval	40 milliseconds. The timer code on the IBM PC/AT is accurate to within 1 millisecond, so a 40 millisecond burst can be timed very accurately.

Allowing only one outstanding buffer reduced the protocol to running "lock-step" (the receiver of data sends a GO, the sender sends data, the receiver sends an OK, followed by a GO for the next buffer). Since the lock-step test incurred one round-trip-delay's worth of overhead per buffer (between transmission of a buffer's last data packet and receipt of an OK for that buffer/GO for the next buffer), a test with two outstanding buffers (providing essentially constant packet transmission) should have resulted in higher throughput.

A second test, this time with two outstanding buffers, was performed, with the above parameters identical save for an increased burst interval of 43 milliseconds. The highest throughput recorded was 1.75 megabits per second. This represents 95% efficiency (5 1990-byte packets every 43 milliseconds gives a maximum theoretical throughput of 1.85 megabits per second). The increase in throughput over a single-outstanding-buffer transmission occurs because, with two outstanding buffers, there is no round-trip-delay lag between buffer transmissions and the sending NETBLT can transmit constantly. Because the P1300 interface can transmit and receive concurrently, no packets were dropped due to collision on the interface.

As mentioned previously, the minimum transmission time for a maximum-sized packet on the Proteon Ring is 8 milliseconds. One would expect, therefore, that the maximum throughput for a double-buffered transmission would occur with a burst interval of 8 milliseconds times 5 packets per burst, or 40 milliseconds. This would allow the sender of data to transmit bursts with no "dead time" in between bursts. Unfortunately, the sender of data must take time to process incoming control messages, which typically forces a 2-3 millisecond gap between bursts, lowering the throughput. With a burst interval of 43 milliseconds, the incoming packets are processed

during the 3 millisecond-per-burst "dead time", making the protocol more efficient.

6. Testing on an Ethernet

The network used in performing this series of tests was a 10 megabit per second Ethernet supporting about 150 hosts. The machines at either end of the NETBLT connection were IBM PC/ATs using 3COM 3C500 network interfaces. As with the Proteon Token Ring, the Ethernet provides high bandwidth with low delay. Unfortunately, the particular Ethernet used for testing (MIT's infamous Subnet 26) is known for being somewhat noisy. In addition, the 3COM 3C500 Ethernet interfaces are relatively unsophisticated, with only a single hardware packet buffer for both transmitting and receiving packets. This gives the interface an annoying tendency to drop packets under heavy load. The combination of these factors made protocol performance analysis somewhat more difficult than on the Proteon Ring.

The fastest single-buffer transmission rate was 1.45 megabits per second, and was achieved using a test case with the following parameters:

transfer size	2-5 million bytes
data packet size	1438 bytes (maximum size excluding protocol headers).
buffer size	14380 bytes
burst size	5 packets
burst interval	30 milliseconds (6.0 milliseconds x 5 packets).

A second test, this one with parameters identical to the first save for number of outstanding buffers (2 instead of 1) resulted in substantially lower throughput (994 kilobits per second), with a large number of packets retransmitted (10%). The retransmissions occurred because the 3COM 3C500 network interface has only one hardware packet buffer and cannot hold a transmitting and receiving packet at the same time. With two outstanding buffers, the sender of data can transmit constantly; this means that when the receiver of data attempts to send a packet, its interface's receive hardware goes

deaf to the network and any packets being transmitted at the time by the sender of data are lost. A symmetrical problem occurs with control messages sent from receiver of data to sender of data, but the number of control messages sent is small enough and the retransmission algorithm redundant enough that little performance degradation occurs due to control message loss.

When the burst interval was lengthened from 30 milliseconds per 5 packet burst to 45 milliseconds per 5 packet burst, a third as many packets were dropped, and throughput climbed accordingly, to 1.12 megabits per second. Presumably, the longer burst interval allowed more dead time between bursts and less likelihood of the receiver of data's interface being deaf to the net while the sender of data was sending a packet. An interesting note is that, when the same test was conducted on a special Ethernet LAN with the only two hosts attached being the two NETBLT machines, no packets were dropped once the burst interval rose above 40 milliseconds/5 packet burst. The improved performance was doubtless due to the absence of extra network traffic.

7. Testing on the Wideband Network

The following section describes results gathered using the Wideband network. The Wideband network is a satellite-based network with ten stations competing for a raw satellite channel bandwidth of 3 megabits per second. Since the various tests resulted in substantial changes to the NETBLT specification and implementation, some of the major changes are described along with the results and problems that forced those changes.

The Wideband network has several characteristics that make it an excellent environment for testing NETBLT. First, it has an extremely long round-trip delay (1.8 seconds). This provides a good test of NETBLT's rate control and multiple-buffering capabilities. NETBLT's rate control allows the packet transmission rate to be regulated independently of the maximum allowable amount of outstanding data, providing flow control as well as very large "windows". NETBLT's multiple-buffering capability enables data to still be transmitted while earlier data are awaiting retransmission and subsequent data are being prepared for transmission. On a network with a long round-trip delay, the alternative "lock-step" approach would require a 1.8 second gap between each buffer transmission, degrading performance.

Another interesting characteristic of the Wideband network is its throughput. Although its raw bandwidth is 3 megabits per second, at the time of these tests fully 2/3 of that was consumed by low-level network overhead and hardware limitations. (A detailed analysis of

the overhead appears at the end of this document.) This reduces the available bandwidth to just over 1 megabit per second. Since the NETBLT implementation can run substantially faster than that, testing over the Wideband net allows us to measure NETBLT's ability to utilize very high percentages of available bandwidth.

Finally, the Wideband net has some interesting packet reorder and delay characteristics that provide a good test of NETBLT's ability to deal with these problems.

Testing progressed in several phases. The first phase involved using source-routed packets in a path from an IBM PC/AT on MIT's Subnet 26, through a BBN Butterfly Gateway, over a T1 link to BBN, onto the Wideband network, back down into a BBN Voice Funnel, and onto ISI's Ethernet to another IBM PC/AT. Testing proceeded fairly slowly, due to gateway software and source-routing bugs. Once a connection was finally established, we recorded a best throughput of approximately 90K bits per second.

Several problems contributed to the low throughput. First, the gateways at either end were forwarding packets onto their respective LANs faster than the IBM PC/AT's could accept them (the 3COM 3C500 interface would not have time to re-enable input before another packet would arrive from the gateway). Even with bursts of size 1, spaced 6 milliseconds apart, the gateways would aggregate groups of packets coming from the same satellite frame, and send them faster than the PC could receive them. The obvious result was many dropped packets, and degraded performance. Also, the half-duplex nature of the 3COM interface caused incoming packets to be dropped when packets were being sent.

The number of packets dropped on the sending NETBLT side due to the long interface re-enable time was reduced by packing as many control messages as possible into a single control packet (rather than placing only one message in a control packet). This reduced the number of control packets transmitted to one per buffer transmission, which the PC was able to handle. In particular, messages of the form OK(n) were combined with messages of the form GO(n + 1), in order to prevent two control packets from arriving too close together to both be received.

Performance degradation from dropped control packets was also minimized by changing to a highly redundant control packet transmission algorithm. Control messages are now stored in a single long-lived packet, with ACKed messages continuously bumped off the head of the packet and new messages added at the tail of the packet. Every time a new message needs to be transmitted, any unACKed old messages are transmitted as well. The sending NETBLT, which receives

these control messages, is tuned to ignore duplicate messages with almost no overhead. This transmission redundancy puts little reliance on the NETBLT control timer, further reducing performance degradation from lost control packets.

Although the effect of dropped packets on the receiving NETBLT could not be completely eliminated, it was reduced somewhat by some changes to the implementation. Data packets from the sending NETBLT are guaranteed to be transmitted by buffer number, lowest number first. In some cases, this allowed the receiving NETBLT to make retransmit-request decisions for a buffer N, if packets for N were expected but none were received at the time packets for a buffer N+M were received. This optimization was somewhat complicated, but improved NETBLT's performance in the face of missing packets. Unfortunately, the dropped-packet problem remained until the NETBLT implementation was ported to a SUN-3 workstation. The SUN is able to handle the incoming packets quite well, dropping only 0.5% of the data packets (as opposed to the PC's 15 - 20%).

Another problem with the Wideband network was its tendency to re-order and delay packets. Dealing with these problems required several changes in the implementation. Previously, the NETBLT implementation was "optimized" to generate retransmit requests as soon as possible, if possible not relying on expiration of a data timer. For instance, when the receiving NETBLT received an LDATA packet for a buffer N, and other packets in buffer N had not arrived, the receiver would immediately generate a RESEND for the missing packets. Similarly, under certain circumstances, the receiver would generate a RESEND for a buffer N if packets for N were expected and had not arrived before packets for a buffer N+M. Obviously, packet-reordering made these "optimizations" generate retransmit requests unnecessarily. In the first case, the implementation was changed to no longer generate a retransmit request on receipt of an LDATA with other packets missing in the buffer. In the second case, a data timer was set with an updated (and presumably more accurate) value, hopefully allowing any re-ordered packets to arrive before timing out and generating a retransmit request.

It is difficult to accommodate Wideband network packet delay in the NETBLT implementation. Packet delays tend to occur in multiples of 600 milliseconds, due to the Wideband network's datagram reservation scheme. A timer value calculation algorithm that used a fixed variance on the order of 600 milliseconds would cause performance degradation when packets were lost. On the other hand, short fixed variance values would not react well to the long delays possible on the Wideband net. Our solution has been to use an adaptive data timer value calculation algorithm. The algorithm maintains an average inter-packet arrival value, and uses that to determine the

data timer value. If the inter-packet arrival time increases, the data timer value will lengthen.

At this point, testing proceeded between NETBLT implementations on a SUN-3 workstation and an IBM PC/AT. The arrival of a Butterfly Gateway at ISI eliminated the need for source-routed packets; some performance improvement was also expected because the Butterfly Gateway is optimized for IP datagram traffic.

In order to put the best Wideband network test results in context, a short analysis follows, showing the best throughput expected on a fully loaded channel. Again, a detailed analysis of the numbers that follow appears at the end of this document.

The best possible datagram rate over the current Wideband configuration is 24,054 bits per channel frame, or 3006 bytes every 21.22 milliseconds. Since the transmission route begins and ends on an Ethernet, the largest amount of data transmissible (after accounting for packet header overhead) is 1438 bytes per packet. This translates to approximately 2 packets per frame. Since we want to avoid overflowing the channel, we should transmit slightly slower than the channel frame rate of 21.2 milliseconds. We therefore came up with a best possible throughput of 2 1438-byte packets every 22 milliseconds, or 1.05 megabits per second.

Because of possible software bugs in either the Butterfly Gateway or the BSAT (gateway-to-earth-station interface), 1438-byte packets were fragmented before transmission over the Wideband network, causing packet delay and poor performance. The best throughput was achieved with the following values:

transfer size	500,000 - 750,000 bytes
data packet size	1432 bytes
buffer size	14320 bytes
burst size	5 packets
burst interval	55 milliseconds

Steady-state throughputs ranged from 926 kilobits per second to 942 kilobits per second, approximately 90% channel utilization. The

amount of data transmitted should have been an order of magnitude higher, in order to get a longer steady-state period; unfortunately at the time we were testing, the Ethernet interface of ISI's Butterfly Gateway would lock up fairly quickly (in 40-60 seconds) at packet rates of approximately 90 per second, forcing a gateway reset. Transmissions therefore had to take less than this amount of time. This problem has reportedly been fixed since the tests were conducted.

In order to test the Wideband network under overload conditions, we attempted several tests at rates of 5 1432-byte packets every 50 milliseconds. At this rate, the Wideband network ground to a halt as four of the ten network BSATs immediately crashed and reset their channel processor nodes. Apparently, the BSATs crash because the ESI (Earth Station Interface), which sends data from the BSAT to the satellite, stops its transmit clock to the BSAT if it runs out of buffer space. The BIO interface connecting BSAT and ESI does not tolerate this clock-stopping, and typically locks up, forcing the channel processor node to reset. A more sophisticated interface, allowing faster transmissions, is being installed in the near future.

8. Future Directions

Some more testing needs to be performed over the Wideband Network in order to get a complete analysis of NETBLT's performance. Once the Butterfly Gateway Ethernet interface lockup problem described earlier has been fixed, we want to perform transmissions of 10 to 50 million bytes to get accurate steady-state throughput results. We also want to run several NETBLT processes in parallel, each tuned to take a fraction of the Wideband Network's available bandwidth. Hopefully, this will demonstrate whether or not burst synchronization across different NETBLT processes will cause network congestion or failure. Once the BIO BSAT-ESI interface is upgraded, we will want to try for higher throughputs, as well as greater hardware stability under overload conditions.

As far as future directions of research into NETBLT, one important area needs to be explored. A series of algorithms need to be developed to allow dynamic selection and control of NETBLT's transmission parameters (burst size, burst interval, and number of outstanding buffers). Ideally, this dynamic control will not require any information from outside sources such as gateways; instead, NETBLT processes will use end-to-end information in order to make transmission rate decisions in the face of noisy channels and network congestion. Some research on dynamic rate control is taking place now, but much more work needs done before the results can be integrated into NETBLT.

I. Wideband Bandwidth Analysis

Although the raw bandwidth of the Wideband Network is 3 megabits per second, currently only about 1 megabit per second of it is available to transmit data. The large amount of overhead is due to the channel control strategy (which uses a fixed-width control subframe based on the maximum number of stations sharing the channel) and the low-performance BIO interface between BBN's BSAT (Butterfly Satellite Interface) and Linkabit's ESI (Earth Station Interface). Higher-performance BSMI interfaces are soon to be installed in all Wideband sites, which should improve the amount of available bandwidth.

Bandwidth on the Wideband network is divided up into frames, each of which has multiple subframes. A frame is 32768 channel symbols, at 2 bits per symbol. One frame is available for transmission every 21.22 milliseconds, giving a raw bandwidth of 65536 bits / 21.22 ms, or 3.081 megabits per second.

Each frame contains two subframes, a control subframe and a data subframe. The control subframe is subdivided into ten slots, one per earth station. Control information takes up 200 symbols per station. Because the communications interface between BSAT and ESI only runs at 2 megabits per second, there must be a padding interval of 1263 symbols between each slot of information, bringing the total control subframe size up to 1463 symbols x 10 stations, or 14630 symbols. The data subframe then has 18138 symbols available. The maximum datagram size is currently expressed as a 14-bit quantity, further dropping the maximum amount of data in a frame to 16384 symbols. After header information is taken into account, this value drops to 16,036 symbols. At 2 bits per symbol, using a 3/4 coding rate, the actual amount of usable data in a frame is 24,054 bits, or approximately 3006 bytes. Thus the theoretical usable bandwidth is 24,054 bits every 21.22 milliseconds, or 1.13 megabits per second. Since the NETBLT implementations are running on Ethernet LANs gatewayed to the Wideband network, the 3006 bytes per channel frame of usable bandwidth translates to two maximum-sized (1500 bytes) Ethernet packets per channel frame, or 1.045 megabits per second.

II. Detailed Proteon Ring LAN Test Results

Following is a table of some of the test results gathered from testing NETBLT between two IBM PC/ATs on a Proteon Token Ring LAN. The table headers have the following definitions:

BS/BI	burst size in packets and burst interval in milliseconds
PSZ	number of bytes in DATA/LDATA packet data segment
BFSZ	number of bytes in NETBLT buffer
XFSZ	number of kilobytes in transfer
NBUFS	number of outstanding buffers
#LOSS	number of data packets lost
#RXM	number of data packets retransmitted
DTMOS	number of data timeouts on receiving end
SPEED	steady-state throughput in megabits per second

BS/BI	PSZ	BFSZ	XFSZ	NBUFS	#LOSS	#RXM	DTMOS	SPEED
5/25	1438	14380	1438	1	0	0	0	1.45
5/25	1438	14380	1438	1	0	0	0	1.45
5/30	1438	14380	1438	1	0	0	0	1.45
5/30	1438	14380	1438	1	0	0	0	1.45
5/35	1438	14380	1438	1	0	0	0	1.40
5/35	1438	14380	1438	1	0	0	0	1.41
5/40	1438	14380	1438	1	0	0	0	1.33
5/40	1438	14380	1438	1	0	0	0	1.33
5/25	1438	14380	1438	2	0	0	0	1.62
5/25	1438	14380	1438	2	0	0	0	1.61
5/30	1438	14380	1438	2	0	0	0	1.60
5/30	1438	14380	1438	2	0	0	0	1.61
5/34	1438	14380	1438	2	0	0	0	1.59
5/35	1438	14380	1438	2	0	0	0	1.58
5/25	1990	19900	1990	1	0	0	0	1.48
5/25	1990	19900	1990	1	0	0	0	1.49
5/30	1990	19900	1990	1	0	0	0	1.48
5/30	1990	19900	1990	1	0	0	0	1.48
5/35	1990	19900	1990	1	0	0	0	1.49
5/35	1990	19900	1990	1	0	0	0	1.48
5/40	1990	19900	1990	1	0	0	0	1.49
5/40	1990	19900	1990	1	0	0	0	1.49
5/45	1990	19900	1990	1	0	0	0	1.45
5/45	1990	19900	1990	1	0	0	0	1.46
5/25	1990	19900	1990	2	0	0	0	1.75
5/25	1990	19900	1990	2	0	0	0	1.75
5/30	1990	19900	1990	2	0	0	0	1.74
5/30	1990	19900	1990	2	0	0	0	1.75
5/35	1990	19900	1990	2	0	0	0	1.74
5/35	1990	19900	1990	2	0	0	0	1.74
5/40	1990	19900	1990	2	0	0	0	1.75
5/40	1990	19900	1990	2	0	0	0	1.74
5/43	1990	19900	1990	2	0	0	0	1.75
5/43	1990	19900	1990	2	0	0	0	1.74
5/43	1990	19900	1990	2	0	0	0	1.75
5/44	1990	19900	1990	2	0	0	0	1.73
5/44	1990	19900	1990	2	0	0	0	1.72
5/45	1990	19900	1990	2	0	0	0	1.70
5/45	1990	19900	1990	2	0	0	0	1.72

III. Detailed Ethernet LAN Testing Results

Following is a table of some of the test results gathered from testing NETBLT between two IBM PC/ATs on an Ethernet LAN. See previous appendix for table header definitions.

BS/BI	PSZ	BFSZ	XFSZ	NBUFS	#LOSS	#RXM	DTMOS	SPEED
5/30	1438	14380	1438	1	9	9	6	1.42
5/30	1438	14380	1438	1	2	2	2	1.45
5/30	1438	14380	1438	1	5	5	4	1.44
5/35	1438	14380	1438	1	7	7	7	1.38
5/35	1438	14380	1438	1	6	6	5	1.38
5/40	1438	14380	1438	1	48	48	44	1.15
5/40	1438	14380	1438	1	50	50	38	1.17
5/40	1438	14380	1438	1	13	13	11	1.28
5/40	1438	14380	1438	1	7	7	5	1.30
5/30	1438	14380	1438	2	206	206	198	0.995
5/30	1438	14380	1438	2	213	213	198	0.994
5/40	1438	14380	1438	2	117	121	129	1.05
5/40	1438	14380	1438	2	178	181	166	0.892
5/40	1438	14380	1438	2	135	138	130	1.03
5/45	1438	14380	1438	2	57	57	52	1.12
5/45	1438	14380	1438	2	97	97	99	1.02
5/45	1438	14380	1438	2	62	62	51	1.09
5/15	512	10240	2048	1	6	6	4	0.909
5/15	512	10240	2048	1	10	11	7	0.907
5/18	512	10240	2048	1	11	11	8	0.891
5/18	512	10240	2048	1	5	5	9	0.906
5/19	512	10240	2048	1	3	3	3	0.905
5/19	512	10240	2048	1	8	8	7	0.898
5/20	512	10240	2048	1	7	7	4	0.876
5/20	512	10240	2048	1	11	12	5	0.871
5/20	512	10240	2048	1	8	9	5	0.874
5/30	512	10240	2048	2	113	116	84	0.599
5/30	512	10240	2048	2	20	20	14	0.661
5/30	512	10240	2048	2	49	50	40	0.638

IV. Detailed Wideband Network Testing Results

Following is a table of some of the test results gathered from testing NETBLT between an IBM PC/AT and a SUN-3 using the Wideband satellite network. See previous appendix for table header definitions.

BS/BI	PSZ	BFSZ	XFSZ	NBUFS	#LOSS	#RXM	SPEED
5/90	1400	14000	500	22	9	10	0.584
5/90	1400	14000	500	22	12	12	0.576
5/90	1400	14000	500	22	3	3	0.591
5/90	1420	14200	500	22	12	12	0.591
5/90	1420	14200	500	22	6	6	0.600
5/90	1430	14300	500	22	9	10	0.600
5/90	1430	14300	500	22	15	15	0.591
5/90	1430	14300	500	22	12	12	0.590
5/90	1432	14320	716	22	13	16	0.591
5/90	1434	14340	717	22	33	147	0.483
5/90	1436	14360	718	22	25	122	0.500
5/90	1436	14360	718	22	25	109	0.512
5/90	1436	14360	718	22	28	153	0.476
5/90	1438	14380	719	22	6	109	0.533
5/80	1432	14320	716	22	56	68	0.673
5/80	1432	14320	716	22	14	14	0.666
5/80	1432	14320	716	22	15	16	0.661
5/60	1432	14320	716	22	19	22	0.856
5/60	1432	14320	716	22	84	95	0.699
5/60	1432	14320	716	22	18	21	0.871
5/60	1432	14320	716	30	38	40	0.837
5/60	1432	14320	716	30	25	26	0.869
5/55	1432	14320	716	22	13	13	0.935
5/55	1432	14320	716	22	25	25	0.926
5/55	1432	14320	716	22	25	25	0.926
5/55	1432	14320	716	22	20	20	0.932
5/55	1432	14320	716	22	17	19	0.934
5/55	1432	14320	716	22	13	14	0.942